

Article

Image-Based Automatic Watermeter Reading under Challenging Environments

Qingqi Hong ¹, Yiwei Ding ¹, Jinpeng Lin ¹, Meihong Wang ¹, Qingyang Wei ², Xianwei Wang ¹ and Ming Zeng ^{1,*}

¹ School of Informatics, Xiamen University, Xiamen 361000, China; hongqq@xmu.edu.cn (Q.H.); dingyiwei@stu.xmu.edu.cn (Y.D.); jplin@stu.xmu.edu.cn (J.L.); wangmh@xmu.edu.cn (M.W.); 24320182203282@stu.xmu.edu.cn (X.W.)

² School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China; weiqy@ustb.edu.cn

* Correspondence: zengming@xmu.edu.cn

Abstract: With the rapid development of artificial intelligence and fifth-generation mobile network technologies, automatic instrument reading has become an increasingly important topic for intelligent sensors in smart cities. We propose a full pipeline to automatically read watermeters based on a single image, using deep learning methods to provide new technical support for an intelligent water meter reading. To handle the various challenging environments where watermeters reside, our pipeline disentangled the task into individual subtasks based on the structures of typical watermeters. These subtasks include component localization, orientation alignment, spatial layout guidance reading, and regression-based pointer reading. The devised algorithms for orientation alignment and spatial layout guidance are tailored to improve the robustness of our neural network. We also collect images of watermeters in real scenes and build a dataset for training and evaluation. Experimental results demonstrate the effectiveness of the proposed method even under challenging environments with varying lighting, occlusions, and different orientations. Thanks to the lightweight algorithms adopted in our pipeline, the system can be easily deployed and fully automated.

Keywords: watermeter reading; automatic method; neural network; deep learning



Citation: Hong, Q.; Ding, Y.; Lin, J.; Wang, M.; Wei, Q.; Wang, X.; Zeng, M. Image-based Automatic Watermeter Reading Under Challenging Environments. *Sensors* **2021**, *21*, 434. <https://doi.org/10.3390/s21020434>

Received: 30 November 2020

Accepted: 4 January 2021

Published: 9 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Automation is widely used to optimize processes and facilitate labor-intensive tasks in our daily life. In the field of smart cities, with the development of technologies of artificial intelligence and fifth-generation mobile networks, as a core part of the intelligent sensors, the technique of automatic device reading has become increasingly critical and technically feasible. Automatic watermeter reading is one such practical and challenging task. In recent years, many methods related to automated meter reading have been proposed to make this work more convenient.

Current automatic watermeter reading methods, however, do not generalize well to typical daily usage scenarios. One kind of current method involves equipping a miniature camera to the watermeter [1–4]. Although effective, this increases expenses because of the additional camera setup required for each watermeter. Another kind of method is sensor-based. These methods [5–7] take an alternative approach, and integrate wireless transceivers into the watermeter. Water flow can be sensed by performing adaptive signal processing on the generated voltage to provide real-time water flow information.

Both types of smart readers methods described above require high switching costs. Despite the existence of smart readers, they are not widespread in many countries, especially in the underdeveloped ones, and it is still manually read on site by the operator who uses the image as proof of reading. Moreover, since there are many images to be evaluated, the traditional manual reading of the watermeter is tedious and error-prone.

To address this dilemma, we propose taking a photo with a mobile phone and automatically analyzing the watermeter reading in this image. Because of the diversity of imaging conditions, especially the potentially challenging environments where the watermeters reside, it is not a trivial work to develop a robust automatic reading system for watermeters. As shown in Figure 1, real images of watermeters come with a variety of challenges. First, the lighting, resolution, and background environment varies across images. Second, the position and rotation angle of the watermeter is unpredictable. Finally, the watermeter is usually covered with dust, and the resolution of the image is low, making it tough to accurately read the watermeter value.



Figure 1. Watermeters under different challenging environments in real life.

To solve these challenges, we leverage the techniques of CNN-based deep learning [8], including object detection [9], classification [10], and regression [11,12], to complete watermeter reading automatically. Our method requires manual taking the watermeter image and can automatically take this reading, which avoids the tedious reading process and ensures the reading's accuracy in a low-cost way, effectively preventing artificial tampering with data and significantly improving efficiency and accuracy.

This paper explores object detection, orientation alignment, spatial layout guidance digit localization, and value regression to implement automatic watermeter reading. We use object detection to extract the position of the watermeter in the input image and crop it out. Then, the orientation alignment algorithm is utilized to adjust the cropped image to the correct reading orientation. Next, we extract the part of the watermeter that contains the reading information (a digit box and several pointer meters). Finally, we utilize the spatial layout guidance algorithm to locate each digit and then read the digits and pointer values to obtain the final reading. Particularly under challenging environments, the devised orientation alignment ensures that the reading task is executed at the correct angle; additionally, the spatial layout guidance algorithm helps us locate each digit accurately. Based on these building blocks, we propose an end-to-end pipeline and train it on a large and challenging environment dataset, yielding a robust automatic reading system for watermeters.

In summary, the contributions of this paper are as follows:

- We propose a robust end-to-end system based on convolutional neural networks for automatic reading of structured watermeter instruments. Our method tailors and combines the latest object detection, feature point location, and novel angle regression techniques.
- We design an orientation alignment algorithm for image correction and propose a spatial layout guidance algorithm to locate digits.
- We carry out a comprehensive experimental analysis that shows that our method effectively meets the challenges of various environmental factors and achieve reliable meter reading performance.

- We build a large-scale watermeter dataset including 9500 training images and 500 test images. To the best of our knowledge, this is the largest watermeter dataset with images taken under different challenging environments. This dataset can further improve the robustness of our automatic readings.

2. Related Work

This section reviews the methods employed for the automatic reading of instruments and introduces the procedures we utilize for automated meter reading, object detection, and text detection.

2.1. Automatic Meter Reading

There are many types of automatic meter reading usage scenarios in real life. The most widely used are pointer reading and digit reading. In terms of pointer reading, the traditional computer vision method combines the binary image subtraction [13] and the Hough transform [14] to estimate the angle of the pointer. However, this method is not robust enough for complex environments, such as various backgrounds and lighting. Zuo et al. [15] improved the existing Mask-RCNN approach [16], classifying the type of pointer meters while predicting the pointer binary mask and then calculating the readings of a pointer table according to the angle of the pointer. However, this method is designed for a specific environment, which reduces the method's application scope in a real-world environment. As for digit reading, Anis et al. [17] propose recognizing digital meter reading based on the Horizontal and Vertical Binary (HVB) patterns. But the digit numbers they process are complete and static, which is not suitable for rolling digit meter reading in a watermeter. Laroca et al. [18] employs the Fast-YOLO object detector for components detection and evaluates three different CNN-based approaches for components recognition. They regarded the reading of the rolling digit as the goal of future work and did not propose a solution. Many researchers have concentrated on improving digit recognition algorithms [19] or classifiers [20] to achieve higher precision in digit recognition. Although previous mentioned digit recognition methods [21,22] have reached very high accuracy, they are not suitable for the watermeter reading task because the digit will roll with the volume of water. For example, for a value of 0.5, the digit turns between 0 and 1 and does not concentrate on 0 or 1. We use the probability distribution matching to solve this problem, the details of which will be introduced in Section 3.4.

2.2. Object Detection

To read a meter, we must locate the digits that need to be read. We use an object detector to predict the positions of the digits that we need to read. Some recent approaches exploited the vertical and horizontal pixel projections histograms [23] for object detection. Other methods took advantage of prior knowledge, such as object position or its colors [24]. These techniques' inevitable shortcoming is that they might not work on all meter types, and the color information might not be stable when the illumination changes. Therefore, we utilize a Convolution Neural Network (CNN)-based object detector to locate the water meter and its inner components. Common CNN-based object detection algorithms can be divided into two categories: two-stage detection algorithms, and one-stage detection algorithms. The former divides the detection problem into two stages. The first stage generates candidate regions, and the second stage classifies candidate regions. The most representative two-stage object detector is the R-CNN [25] series, including fast R-CNN [26], faster R-CNN [27], R-FCN [28], and Libra R-CNN [29]. Meanwhile, one-stage detection algorithms include SSD [30,31] and YOLO [32–34], and do not require the region proposal stage; instead, these algorithms directly generate the category probability and position coordinate value of the object. The two-stage algorithm is accurate and the one-stage algorithm is lightweight. After several version updates, YOLO3 [34] not only reached higher accuracy but also maintained a high running speed. Therefore, we leverage YOLO3 as our watermeter detection model.

2.3. Text Detection

The major trend in scene text detection before the emergence of deep learning was bottom-up, in which case handcrafted features were used most often, such as SWT [35] or MSER [36] as a basic component, but those algorithm failed with blurring and perspective distortions. The current widely used text detectors are as follows: the regression-based text detectors [37,38] adopt object detection methods to find the position of words; the segmentation-based text detectors [39,40] aim to find the pixel-level text area and detect the text by estimating the word boundary area; and character-level text detectors [41,42] detect the text area by exploring each character and the affinity between characters. A major drawback of these techniques is that their results are susceptible to non-text lines.

3. Proposed Method

3.1. Reading Rule of Mechanical Watermeters

Figure 2 depicts a typical mechanical watermeter. It is composed of structured digit panels and corresponding units. Although the structures of watermeters produced by different companies are not the same, they share similar panel layouts and reading rules. Therefore, an automatic reading watermeter method based on the divide-and-parse methodology can be easily adapted to various watermeters with similar structures. Figure 2 also illustrates the reading rule for a typical watermeter. The green box contains the digit box's value (reading this is called digit reading), and the blue box contains the pointer's value (reading this is called pointer reading). The red arrows point to the corresponding units. A weighted sum of these values results in the final water usage reading.

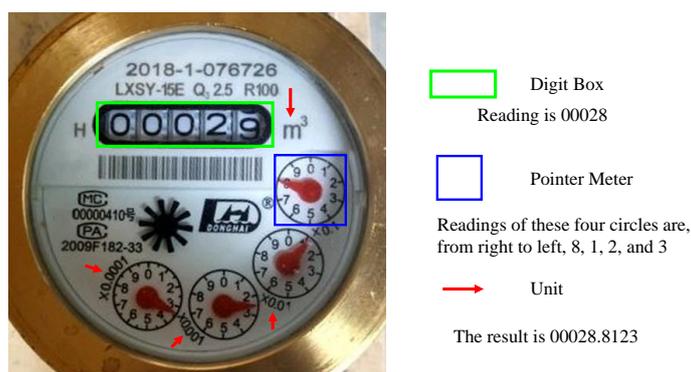


Figure 2. Schematic diagram of the watermeter. The weighted sum of the values of the digit box and the pointer meter indicates the water consumption.

3.2. Overview

According to the reading rule of the mechanical watermeters, and considering the potentially challenging environments in which watermeters are located, we split the reading task into individual subtasks. Figure 3 shows the full pipeline of our system framework. The whole pipeline consists of the following parts: watermeter detection, orientation alignment, digit reading with spatial layout guidance and pointer reading.

Our pipeline takes a watermeter image I as input and then it outputs the corresponding watermeter value. First, the watermeter detection model M_1 detects the position O_1 of the watermeter and then obtain the cropped watermeter image I_2 . Second, we adopt the orientation alignment module M_2 to rotate I_2 ; this is followed by component localization with component detection module M_3 , leading to one bounding box I_5 for the digit box and the four bounding boxes I_6 for pointer meters. Then, we design the keypoint localization model M_4 to localize and separate each digit in I_5 . Finally, we can read the values O_5 and O_6 by the digit reading model M_5 and the pointer reading model M_6 , respectively. The final prediction V is the sum of O_5 and O_6 .

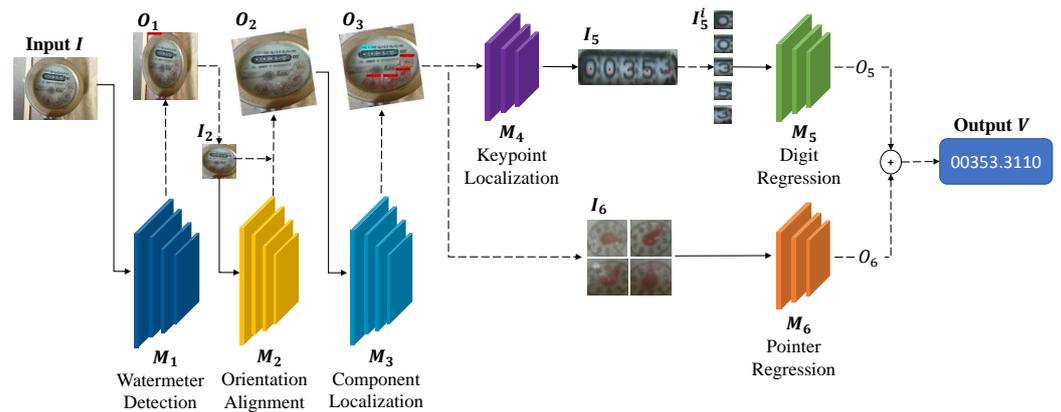


Figure 3. The pipeline of our framework. Our model takes a watermeter image I as input and outputs the corresponding value V . First, the detection model M_1 detects the position O_1 of the watermeter, resulting in the cropped watermeter image I_2 . Second, the orientation alignment module M_2 is adopted to get the aligned image patch O_2 . Then, the component localization module M_3 predicts and locates the digit box I_5 and the regions of pointer meters I_6 . Next, the keypoint localization model M_4 is introduced to localize and separate each digit in I_5 . Finally, the values O_5 and O_6 are obtained by the digit reading model M_5 and the pointer reading model M_6 , respectively. The final prediction V is the sum of O_5 and O_6 .

3.3. Watermeter Detection and Rotation Corrected Component Localization

3.3.1. Watermeter Detection

To accurately and rapidly determine the position of the watermeter, we adopt YOLO3 [34] for one-class (only the watermeter) object detection. Taking an image I as input, YOLO3 outputs the position $(x, y, w, h)_m$ of the watermeter. The position information includes the center $(x, y)_m$, the width (w), and the height (h), where m represents the subscript of detection results.

3.3.2. Orientation Alignment

Because of the 6-DOF transformation of the camera viewpoint, the cropped region I_2 has a perspective transformation. As a result, direct positioning and reading caused problems: as shown in Figure 4, the positioning of the number box is offset (Figure 4b) and the reading is incorrect (Figure 4c).

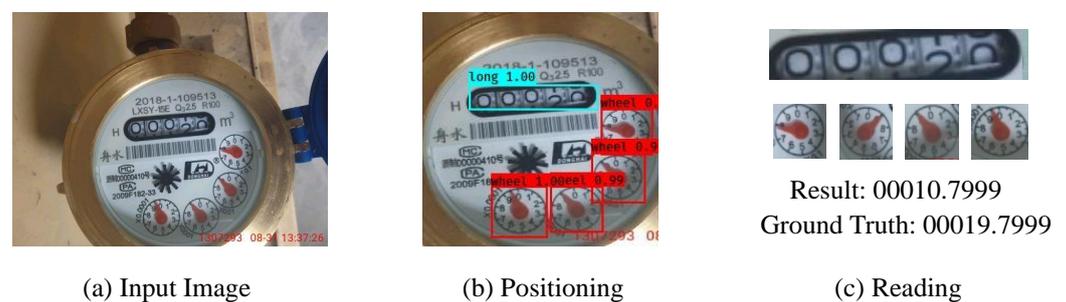


Figure 4. Failure case of incorrect reading caused by the inaccurate localization for the digit box.

Therefore, we propose an orientation alignment network to adjust the reading angle. Our method predicts an angle of in-plane rotation to correct the orientation of I_2 . Although the transformation is actually perspective, in practice the simplification of in-plane rotation is sufficient to account for this variation and achieve satisfactory results. More concretely, we did not directly regress the rotation angle because the angle is periodic. For example, -20° and 340° correspond to the same angle, which is ambiguous. To elim-

inate this ambiguity, we regress the *sin* and *cos* values of a given angle. Hence, the loss function can be formulated as follows:

$$L_{angle} = \|P_{sin} - \sin(\theta)\|^2 + \|P_{cos} - \cos(\theta)\|^2, \quad (1)$$

where P_{sin} and P_{cos} is the output of \mathbb{M}_2 . $\theta \in [-\pi, \pi]$ denotes the ground truth angle.

3.3.3. Component Localization

We use another YOLO3 detection model \mathbb{M}_3 for two-classes (the digit box and the pointer meters) object detection. Using the image O_2 as the input, \mathbb{M}_3 output the position of the digit box and pointer meters, denotes as $(x, y, w, h)_d$ and $\{(x, y, w, h)_p^j\}_{j=1}^4$.

3.4. Regression-Based Digit Reading with Spatial Layout Guidance

3.4.1. Spatial Layout Guidance for Digit Localization

Because of the low image quality caused by bad environments, straightforward methods like uniform character segmentation and Optical Character Recognition (OCR) text detection may fail to locate and read each digit accurately. We have explored two attempts:

1. Given the detected digital region, we uniformly separate each digit and then predict the value for each digit using regression.
2. Directly leverage an off-the-shelf OCR module to recognize the digits.

However, neither of these methods accurately locates the position of each digit. The disadvantages of these straightforward methods are illustrated in Figure 5. The first method relied too heavily on the results of the component localization model \mathbb{M}_3 . If the results of the \mathbb{M}_3 are not accurate enough, incomplete digits will be generated. Meanwhile, the second method often fails when recognizing a rolling digit, which is a common occurrence in watermeters.

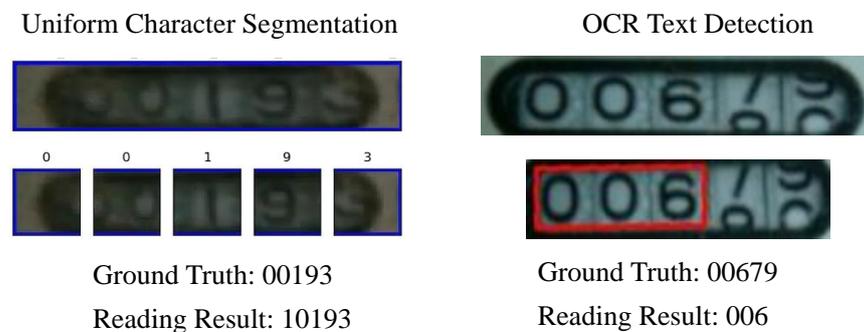


Figure 5. Uniform character segmentation and character detection. The segmentation result on the left is inaccurate, and the digits on the right are not fully detected; both lead to wrong readings.

To address the problem of digital localization, we recast the problem as keypoint localization. We locate the positions of digits $(x_i, y_i) \ i = (1, 2, \dots, 5)$. Because of the low image quality and the transition state between two consecutive digits, the text is not clear enough to robustly localize the digits. We thus utilize the linear spatial layout of the text region as prior information to constrain the keypoint localization. Therefore, we require that the predicted localization of each digit is colinear and equidistant (as illustrated in Figure 6), which could be formulated as neighboring offsets of predicted positions are almost the same. Hence, the loss function is as follows:

$$L_{keypoint} = \sum_{i=1}^N \|P_i - \hat{P}_i\|^2 + \sum_{i=1}^{N-1} \|\Delta \hat{P}_i - \Delta \hat{P}_{i-1}\|^2, \quad (2)$$

where P_i denotes the ground truth of the 2D coordinate of the i th digit position and \hat{P} denotes the predicted coordinate, and $\Delta\hat{P}_i = \hat{P}_{i+1} - \hat{P}_i$ denotes the spatial offset of nearby digits.

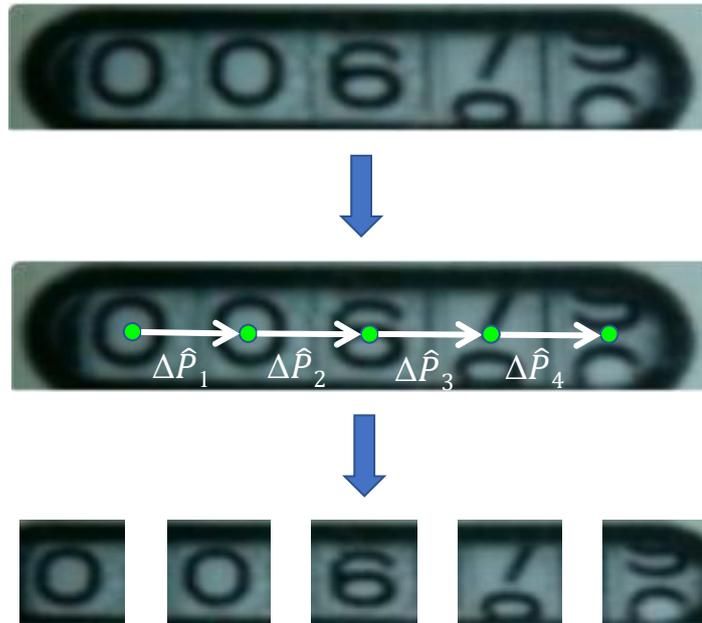


Figure 6. Keypoint localization method with spatial layout guidance. The five green points are on the same line and are equidistant.

3.4.2. Digit Reading

After keypoint localization, we crop the digit box images into five parts according to regressed coordinates. For the ground truth of 9.5, as shown in the far-right digit in Figure 6, the digit appeared as the bottom half of 9 and the top half of 0. A straightforward way to predict the digit value is to regress the value between [0,10] by using Mean Square Error(MSE). But the penalty is different when the model output 0.0 and 9.0 with MSE loss $(9.5 - 0)^2$ vs. $(9.5 - 9.0)^2$, which will provide the wrong update information to the model. This situation is caused by the value jump from 9 to 0. To eliminate this effect, we formulate this task as a Circle Probability Distribution (CPD) prediction problem. Specifically, as shown in Figure 7, we use a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ to calculate probabilities for every discrete integer (ranging from 0 to 9 by step size of 1). Given μ as the ground truth, we set $\sigma = 0.05$ in our experiment, and the optimal model should predict a Gaussian CPD centered at μ .

We use a CNN module denoted as \mathbb{M}_5 for the digit reading. Given input images I_5^i , we can sample ten probabilities $\{p_i\}_{i=0}^9$ as noted earlier, and then \mathbb{M}_5 output ten probabilities $\{\hat{p}_i\}_{i=0}^9$. Categorical cross-entropy loss is introduced to fit the Gaussian CPD, as follows:

$$L_{digit} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=0}^9 p_i^j \log \hat{p}_i^j, \quad (3)$$

where N denotes the training sample number. We assume that the maximum probability (the crest of CPD) locates between indices of I_1 and I_2 with the top-2 predicted probabilities in $\{\hat{p}_0, \hat{p}_1, \dots, \hat{p}_9\}$. We use $\min(I_1, I_2)$ as the final output O_5 according to the watermeter reading rule.

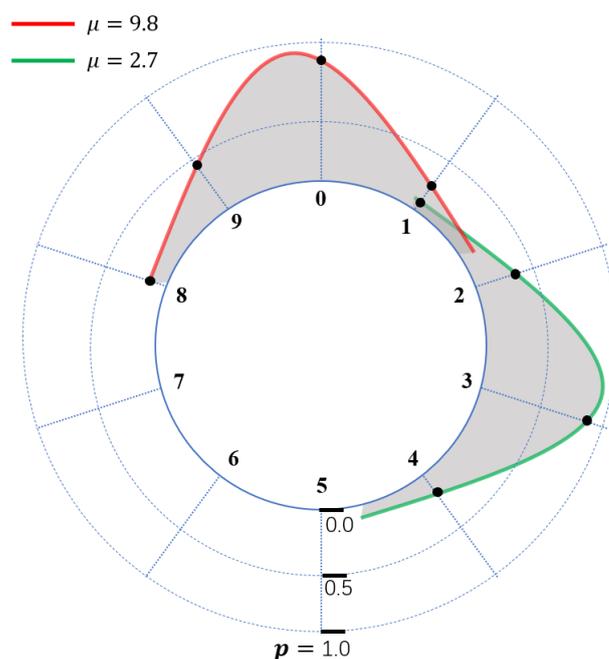


Figure 7. Gaussian Circle Probability Distribution (CPD) centered at different μ values. For each discrete integer in the range $[0,9]$, we sample the corresponding probability (the probability at the black dot) to construct the final ground truth.

3.5. Regression-Based Pointer Reading

To read the value of pointer meters, we first crop O_2 using $(x, y, w, h)_p$ and obtain image I_6 . The rotation angle of the pointer could infer the pointer's value. Because the pointer in the dial meter is discriminative, it is a more natural problem for the neural network to learn the direction of the pointer. To estimate the angle of the pointer, as in Section 3.3, we trained the orientation alignment model to regress the \cos and \sin values of the pointer angle. We use the same regression loss to train the pointer reading model.

4. Experiments

In this section, we first describe the data preparation and the evaluation metrics of our experiments. Then we verify the effectiveness of our proposed method from three aspects: (1) We conduct quantitative and qualitative experiments to demonstrate the performance of our key modules. (2) To evaluate the contribution of each module, we designed ablation studies. (3) Due to the pipeline is designed to be applied to the real world, we also validate our pipeline's robustness under different challenging environments.

4.1. Experiment Setup

4.1.1. Data Preparation

Compared with the widely used datasets in deep learning such as PASCAL VOC [43], COCO [44] and ImageNet [45], to the best of our knowledge, there is no public watermeter dataset with reading annotations. To foster the training for watermeter reading, we collected watermeter images from real life and web crawlers. We hired people to the houses where people actually live and collect watermeter original data by manually taking pictures. After that, we hired people with labeling experience to label the training data through VIA [46] (a simple and powerful manual image annotation tool). The watermeter images in the resulting dataset contain a variety of angles, colors, lighting, resolutions, and background scenes, etc. We randomly divide the collected data set into training data and test data at a ratio of 95% and 5%.

As shown in Figure 3, six models need to be trained in our system, and they are executed sequentially. The latter model depends on the previously trained model. Therefore, we annotate our training data progressively. As shown in Figure 8a, we first annotate

the position of the watermeter at the original collected images. Model M_1 is trained on these annotated images, and the trained model is used to detect watermeter on all original images. Detected watermeters are cropped out, and we can obtain images like Figure 8b. Secondly, we annotate the line segment with direction at cropped watermeter. Then, we calculate the angle based on the annotated line segment and then use the \sin and \cos values of the angle to supervise the training of M_2 . Trained M_2 is then used to correct the orientation of all training data. Thirdly, we annotate the bounding boxes and actual values of digits and pointers, as shown in Figure 8c. With annotated position supervision, we train another detection model M_4 . Trained M_4 is then used to crop out digits and pointers. Finally, M_6 is trained with cropped pointer image and annotated value. M_5 is trained with cropped digits image and annotated digit value. Furthermore, five-digit center points in the digit box, as shown in Figure 8d, are annotated to guide the center localization.

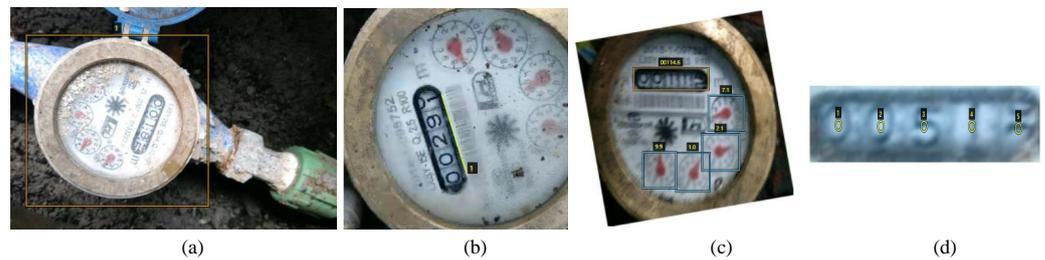


Figure 8. Examples of data annotations required to train different models. (a) annotates the position of the water meter, (b) annotates the direction of watermeter, (c) annotates the position and value of the inner components, and (d) annotates the digit box's center points.

4.1.2. Implementation Details

The backbone of orientation alignment M_2 , spatial layout guidance M_4 and pointer regression M_6 are modified versions of the ResNet-50 [47] excluding the average pooling layer. We initialized all models with pretrained weights (YOLO3 pretrained on the COCO [44] dataset for object detection and ResNet pretrained on the ImageNet [45] dataset for image recognition). For each model, we split the training process into two stages. We update only the last layer of the model for the first stage and then update all layers together for the second stage. We use the Adam [48] optimizer and a learning rate of 10^{-4} are employed for optimization. Input image sizes for the six models are (416×416) , (416×416) , (224×224) , (224×224) , (32×32) and (224×224) .

4.1.3. Evaluation Metrics

We used three evaluation metrics in the experiments: angle error, digit error and pointer error. The output of digit is an integer, so we can judge it directly. The output of pointer is a decimal, and it is judged by whether its value is within ± 0.5 of the ground truth. The calculation formulas for other evaluation metrics are as follows:

$$Angle_{error} = \left\| Angle_{truth} - \hat{Angle}_{pred} \right\|, \quad (4)$$

$$Digit_{error} = \frac{Digits_{wrong}}{Digits_{all}}, \quad (5)$$

$$Pointer_{error} = \frac{Pointer_{wrong}}{Pointer_{all}}, \quad (6)$$

where $Angle_{truth}$ denotes the ground truth rotation angle of the watermeter, \hat{Angle}_{pred} denotes the predicted rotation angle of the watermeter. $Digits_{wrong}$ denotes the number of incorrectly predicted digits, $Digits_{all}$ denotes the total number of digits. And $Pointer_{wrong}$ denotes the number of incorrectly predicted pointers, $Pointer_{all}$ denotes the total number of pointers.

4.2. Performance Evaluation for Key Modules

4.2.1. Orientation Alignment

We eliminate the ambiguity of the angle periodicity by predicting the \sin and \cos values of the angle. Then we calculate the angle according to these values. To test the performance of our orientation alignment model M_2 in different rotation intervals, we first correct the original data according to the annotation pair and then rotate it randomly. The rotation angle meets the uniform distribution of different intervals. The results of each interval are given in Table 1 and the qualitative results are shown in Figure 9. The average error between the actual angle and recognition angle is less than 1 degree. The experimental result illustrates that the method finely amends the error caused by the slanted image.

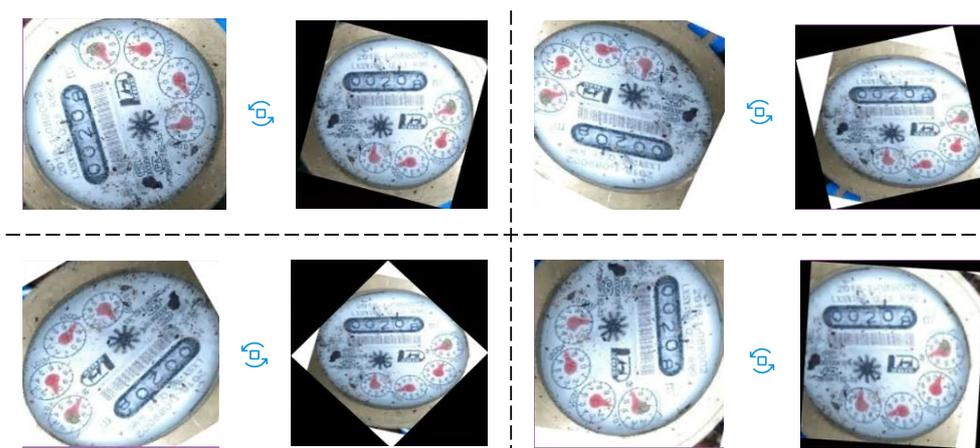


Figure 9. Qualitative results of orientation alignment. Regardless of how much the image angle shifted, the orientation alignment module adjusts it correctly.

Table 1. The quantitative results for the orientation alignment module. The evaluation metric is the angle error.

Angle Distribution	Max Ang. Err.	Min Ang. Err.	Average Ang. Err.
$U(-10^\circ, 10^\circ)$	3.488°	0.001°	0.688°
$U(-20^\circ, 20^\circ)$	3.736°	0.001°	0.713°
$U(-30^\circ, 30^\circ)$	4.690°	0.004°	0.700°
$U(-40^\circ, 40^\circ)$	4.363°	0.003°	0.822°
$U(-50^\circ, 50^\circ)$	3.131°	0.001°	0.735°
$U(-60^\circ, 60^\circ)$	4.791°	0.001°	0.753°
$U(-70^\circ, 70^\circ)$	3.982°	0.001°	0.750°
$U(-80^\circ, 80^\circ)$	3.585°	0.003°	0.726°
$U(-90^\circ, 90^\circ)$	4.026°	0.001°	0.709°

In addition, the orientation alignment module also plays a key role in guiding the recognition rate and accuracy of the digit box. For this purpose, we conducted tests on the test dataset (see the results in Table 2).

Table 2. Effectiveness of orientation alignment on the recognition of the digital box. Average IOU and AP are used as evaluation metrics for detecting digit.

Orientation Alignment	Average IOU	AP@0.5
✓	0.51 0.92	42.11 98.92

4.2.2. Spatial Layout Guidance for Digit Localization

We determine that M_3 is able to identify the position of the digit box with high precision. Therefore, we directly utilize the uniform character segmentation method to divide each digit. After testing, however, we found that if the position identified by M_3 is not accurate, the segmentation result will be poor (Figure 10 Case 1 and Case 2 in the second column). Therefore, to decouple the digital positioning from the previous step, we utilize the OCR text detector CRAFT [41] to detect each digit. CRAFT is not accurate enough, however, when identifying a rolling digit (Figure 10 Case 3 and Case 4 in the third column). We thus propose the use of spatial guidance before solving this problem. The visual comparison of these methods are shown in Figure 10.

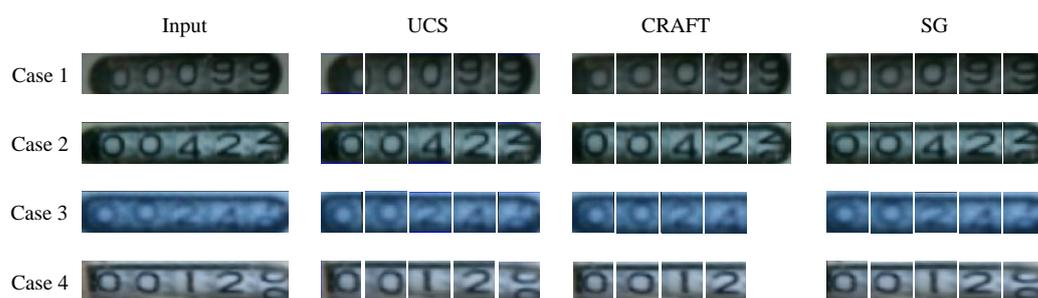


Figure 10. Comparison of segmentation methods. UCS is the uniform character segmentation, CRAFT is the OCR text detector, and SG is the spatial guidance method we proposed.

Besides, to verify the digit localization module's stability, we expanded the test dataset four times by data augmentation methods (e.g., random rotation, scaling, and color transformation) for quantitative experiments. After data augmentation, the difficulty increases, which will inevitably lead to an increase in the error rate. However, the error rate of SG grew slightly, while others' error rates grew significantly. The results are shown in Table 3. We use the error growth rate (error growth rate is the grew digit error ratio the original digit error) as an evaluation metric to make a more intuitive comparison. The tilt of the angle increases the error rate of UCS (digit error increased by 2.32%, with an error growth rate of 27%); the recognition rate of CRAFT is worse due to the color transformation and thus increases the error rate of readings (digit error increased by 1.04%, with an error growth rate of 18%); while the error rate of SG increase in a small range (digit error increased by 0.25%, with an error growth rate of 6%), which means SG stability in to various environments.

Table 3. Ablation study on the test dataset. “*” represents the test dataset with data augmentation. “Basic” means our baseline network. OA denotes orientation alignment, UCS denotes uniform character segmentation, and SG denotes spatial guidance. The evaluation metrics are the digit error and the error growth rate.

Approach	Basic	OA	UCS	CRAFT	SG	Digit Err.
Basic	✓					24.32%
Basic + OA	✓	✓				13.20%
Basic + OA + UCS	✓	✓	✓			8.52%
Basic + OA + UCS + CRAFT	✓	✓	✓	✓		5.72%
Basic + OA + SG	✓	✓			✓	3.79%
* Basic	✓					35.86% (+47%)
* Basic + OA	✓	✓				17.76% (+34%)
* Basic + OA + UCS	✓	✓	✓			10.84 (+27%)
* Basic + OA + UCS + CRAFT	✓	✓	✓	✓		6.76% (+18%)
* Basic + OA + SG	✓	✓			✓	4.04% (+6%)

4.3. Ablation Studies

To understand the role of the orientation alignment and the proposed spatial layout guidance, and to test the effectiveness of each proposed module, we performed ablation studies on our test dataset.

4.3.1. Effectiveness of Orientation Alignment

We first explore the effectiveness of the proposed orientation alignment module. To validate the effectiveness of orientation alignment, we removed this module and analyzed the impact on performance. Table 3 shows the effect of orientation alignment on automatic reading. After adding the orientation alignment module, the reading accuracy is significantly improved. This demonstrates that the orientation alignment module is critical for the automatic reading.

4.3.2. Effectiveness of Spatial Guidance

In Section 4.2, we proved that the spatial guidance method could cope with inaccurate M_3 results and rolling digits. We further compare the influence of the presence or absence of spatial guidance on the reading results. It was evident that each component significantly improves our results, and the spatial guidance method plays an essential role in automatic reading.

A visual comparison is provided in Figure 11. Basic reading failed because the reading is conducted at the wrong orientation (Figure 11b). UCS may have segmented incorrectly, resulting in erroneous readings (Figure 11d). Meanwhile, CRAFT may have made incomplete detections, resulting in incorrect readings (Figure 11e). OA could provide the correct angle for the reading (Figure 11c) and SG accurately segment the digits in the digit box (Figure 11f); notably, their aggregation could generate high-quality reading results.

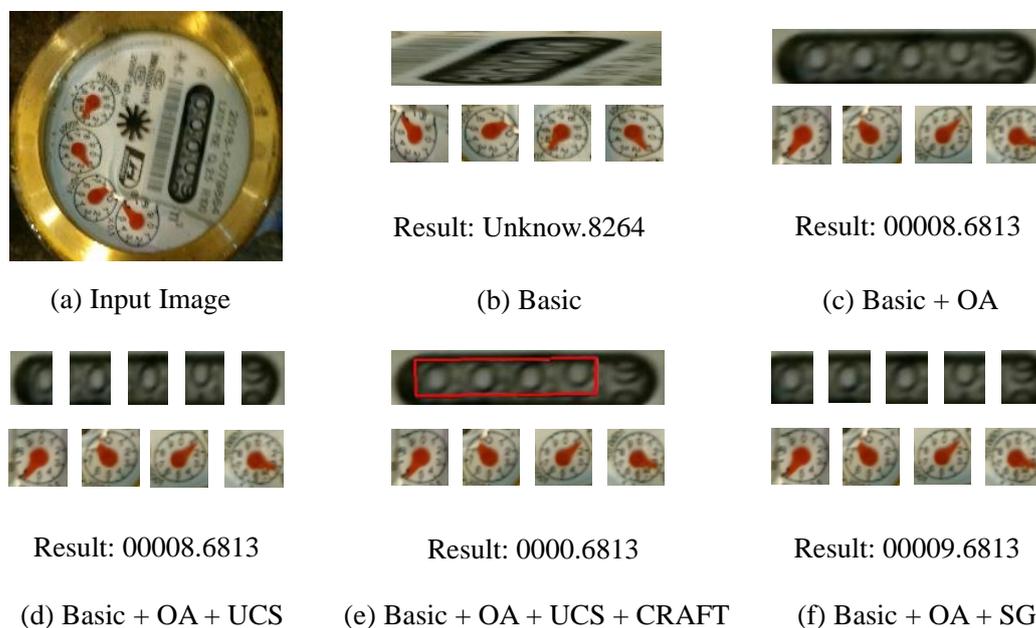


Figure 11. Visual comparison of different methods. “Basic+” represents that we add different components to the baseline to the baseline network to read the watermeter automatically.

4.4. System Performance

This section provides the results of the system’s performance test and briefly describes the deployment of the system.

4.4.1. Robustness to Challenging Environments

To further validate our pipeline’s robustness, we subdivided the test dataset into three major categories based on cleanliness, lighting conditions, and image clarity (see Figure 12). The cleanliness is subdivided into normal and dirty; the lighting environment is subdivided into normal, bright and dark. The image clarity is subdivided into the original sharpness, and the sharpness is reduced twice and three times. There are a total of seven sub-categories, each with 100 pictures. We use the digit error and the pointer error as evaluation metrics, the experimental results are shown in Table 4.



Figure 12. The test dataset images with different cleanliness, lighting and clarity.

Table 4. The quantitative experiments for the pipeline’s robustness. “Down $\times n$ ” means zoom out n times and then zoom in to the original image. The evaluation metrics are the digit error and the pointer error.

		Digit Err.		Pointer Err.	
		Base	Base + OA + SG	Base	Base + OA + SG
Cleanness	Normal	15.0%	3.4%	1.0%	1.0%
	Dirty	19.2%	3.6%	7.0%	3.0%
Lighting	Normal	11.4%	3.4%	2.0%	1.0%
	Bright	13.9%	3.6%	4.0%	2.0%
	Dark	14.2%	3.8%	4.0%	2.0%
Clarity	Normal	13.8%	2.0%	1.0%	0.0%
	Down $\times 2$	14.6%	2.2%	2.0%	2.0%
	Down $\times 3$	16.4%	4.0%	7.0%	3.0%

Experimental results show that our proposed pipeline has the lowest error rate in various environments. Figure 13 shows some of these results: these watermeters had different angles and perspectives; some watermeters were located in dark or light-reflective environments and some watermeters were blurry and covered with dust. Under these various challenging environments, our method achieved satisfactory results.



Figure 13. Performance test of the complete system: the first column is the input, the second to fourth columns are the intermediate results, and the last column is the final reading result.

4.4.2. System Deployment

Based on the proposed method, we developed and deployed an online system for automatic meterwater reading. The system is constructed with Flask, a Python micro-framework to provide API services. All the models are preloaded and run on a PC with an

i7 CPU and a single Nvidia GTX 1080 graphics card. The average time to infer an image is less than 300 ms. The visual interface of the system is shown in Figure 14.

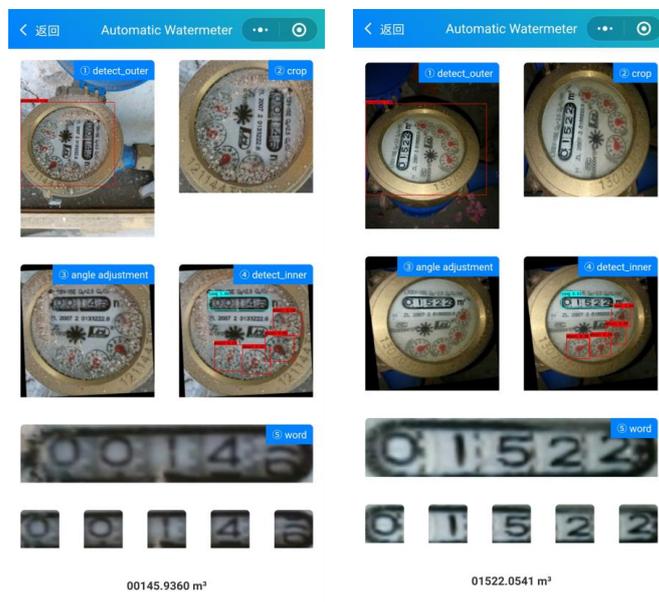


Figure 14. The user interface of the system. Just click to upload the image to complete the meter reading.

4.4.3. Failure Case

Our system is able to deal with various challenging environments, including darkness, blur, glare, dirt, and distant capture, etc. The network, however, could not handle some situations. As shown in Figure 15, when the input image is covered by foreign objects, our system fails to make a reading.

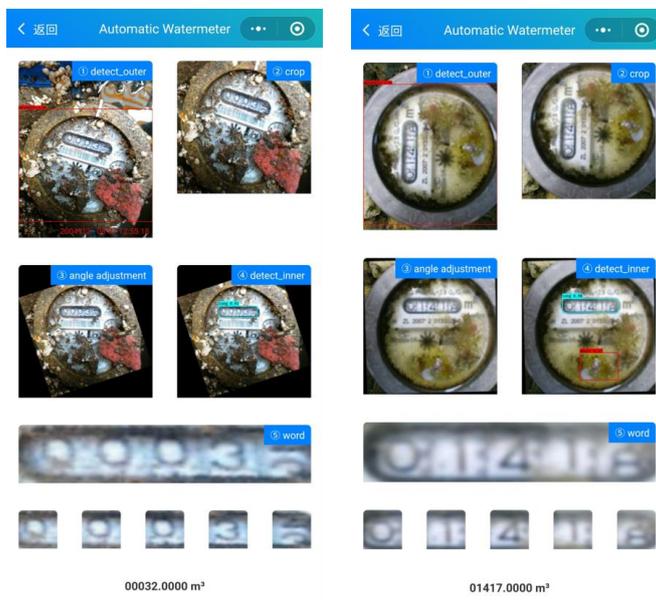


Figure 15. Failure cases. The pointer is partially obscured and cannot be recognized, and therefore cannot be read.

5. Conclusions

In this paper, we propose a fully automatic system for watermeter reading. It is based on end-to-end CNN, including watermeter detection and rotation corrected component localization, regression-based digit reading with spatial layout guidance, and pointer

reading. We construct a new watermeter dataset containing images obtained in various challenging environments for system training and testing. Extensive experiments prove that the orientation alignment module can effectively improve the digit box and pointer detection accuracy; the spatial guidance module can effectively improve the digit box's reading accuracy. Besides, we conducted quantitative experiments to verify the orientation alignment module's efficiency and the spatial guidance module. Moreover, we designed ablation experiments to verify each module's effectiveness in the whole pipeline. To sum up, our method can successfully automatically read water meters under challenging environments with high accuracy, which meets the practical application's requirements.

In future work, we will further improve the accuracy of automatic readings in the actual application process to reduce failure cases. For foreign object coverage, the system will notify the staff to be manual clean; for others, the system will collect these images as part of our dataset. Expanding the real-world environment dataset, we will continue to finetune our model on the expanded data to improve our pipeline's robustness and reduce the error rate.

Author Contributions: Conceptualization, Q.W., M.Z. and M.W.; Investigation, Q.H., M.W. and Q.W.; Methodology, Q.H., M.Z., Y.D., J.L. and M.W.; Project administration, Q.H., Y.D. and M.Z.; Software, Y.D., J.L. and X.W.; Validation, Y.D., J.L. and X.W.; Visualization, Y.D. and X.W.; Writing—original draft, Q.H., Y.D., J.L., X.W. and M.Z.; Writing—review & editing, Q.H., M.W., Q.W. and M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by NSFC (No. 62072382, 61402387, 11975044, 61502402), Guiding Project of Fujian Province, China (No. 2018H0037), Fundamental Research Funds for the Central Universities, China (No.20720190003, 20720180073), and Major Science and Technology Project of Xiamen, China (No. 3502Z20191020).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Goldberg, B.G.; Messinger, G. Remote Meter Reader Using a Network Sensor System and Protocol. U.S. Patent 8,144,027, 27 March 2012.
- Jiale, H.; En, L.; Bingjie, T.; Ming, L. Reading recognition method of analog measuring instruments based on improved hough transform. In Proceedings of the IEEE 2011 10th International Conference on Electronic Measurement & Instruments, Chengdu, China, 16–19 August 2011; IEEE: Piscataway, NJ, USA, 2011; Volume 3, pp. 337–340.
- Wang, J.; Huang, J.; Cheng, R. Automatic Reading System for Analog Instruments Based on Computer Vision and Inspection Robot for Power Plant. In Proceedings of the 2018 10th International Conference on Modelling, Identification and Control (ICMIC), Guiyang, China, 2–4 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–6.
- Zheng, W.; Yin, H.; Wang, A.; Fu, P.; Liu, B. Development of an automatic reading method and software for pointer instruments. In Proceedings of the 2017 First International Conference on Electronics Instrumentation & Information Systems (EIIS), Harbin, China, 3–5 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6.
- Gastouniotis, C.S.; Bandeira, N.; Wilson, K.C. Automated Remote Water Meter Readout System. U.S. Patent 4,940,976, 10 July 1990.
- Mudumbe, M.J.; Abu-Mahfouz, A.M. Smart water meter system for user-centric consumption measurement. In Proceedings of the 2015 IEEE 13th International Conference on Industrial Informatics (INDIN), Cambridge, UK, 22–24 July 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 993–998.
- Li, X.J.; Chong, P.H.J. Design and Implementation of a Self-Powered Smart Water Meter. *Sensors* **2019**, *19*, 4177. [[CrossRef](#)] [[PubMed](#)]
- LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [[CrossRef](#)]
- He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [[CrossRef](#)] [[PubMed](#)]
- Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
- Cohen, G.; Afshar, S.; Tapson, J.; Van Schaik, A. EMNIST: Extending MNIST to handwritten letters. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 2921–2926.
- Deng, L. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Process. Mag.* **2012**, *29*, 141–142. [[CrossRef](#)]

13. Jian, S.; Dong, Z.; Jianguo, H. Design of remote meter reading method for pointer type chemical instruments. *Process Autom. Instrum.* **2014**, *35*, 77–79.
14. Gang, N.; Bin, Y. Pointer instrument image recognition based on priori characteristics of instrument structure. *Electron. Sci. Technol.* **2013**, *26*, 10–12.
15. Zuo, L.; He, P.; Zhang, C.; Zhang, Z. A Robust Approach to Reading Recognition of Pointer Meters Based on Improved Mask-RCNN. *Neurocomputing* **2020**, *388*, 90–101. [[CrossRef](#)]
16. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
17. Anis, A.; Khaliluzzaman, M.; Yakub, M.; Chakraborty, N.; Deb, K. Digital electric meter reading recognition based on horizontal and vertical binary pattern. In Proceedings of the 2017 3rd International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 7–9 December 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6.
18. Laroca, R.; Barroso, V.; Diniz, M.A.; Gonçalves, G.R.; Schwartz, W.R.; Menotti, D. Convolutional neural networks for automatic meter reading. *J. Electron. Imaging* **2019**, *28*, 013023.
19. LeCun, Y.; Jackel, L.; Bottou, L.; Brunot, A.; Cortes, C.; Denker, J.; Drucker, H.; Guyon, I.; Muller, U.; Sackinger, E.; et al. Comparison of learning algorithms for handwritten digit recognition. In Proceedings of the International Conference on Artificial Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 60, pp. 53–60.
20. Bottou, L.; Cortes, C.; Denker, J.S.; Drucker, H.; Guyon, I.; Jackel, L.D.; LeCun, Y.; Muller, U.A.; Sackinger, E.; Simard, P.; et al. Comparison of classifier methods: A case study in handwritten digit recognition. In Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3-Conference C: Signal Processing (Cat. No. 94CH3440-5), Jerusalem, Israel, 9–13 October 1994; IEEE: Piscataway, NJ, USA, 1994; Volume 2, pp. 77–82.
21. Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A.Y. Reading digits in natural images with unsupervised feature learning. Reading digits in natural images with unsupervised feature learning. In Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011, Granada, Spain, 15–16 December 2011.
22. Niu, X.X.; Suen, C.Y. A novel hybrid CNN–SVM classifier for recognizing handwritten digits. *Pattern Recognit.* **2012**, *45*, 1318–1325. [[CrossRef](#)]
23. Edward, V.C.P. Support vector machine based automatic electric meter reading system. In Proceedings of the 2013 IEEE International Conference on Computational Intelligence and Computing Research, Enathi, India, 26–28 December 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 1–5.
24. Elrefaei, L.A.; Bajaber, A.; Natheir, S.; AbuSanab, N.; Bazi, M. Automatic electricity meter reading based on image processing. In Proceedings of the 2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), Amman, Jordan, 3–5 November 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1–5.
25. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
26. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
27. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
28. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 379–387.
29. Pang, J.; Chen, K.; Shi, J.; Feng, H.; Ouyang, W.; Lin, D. Libra r-cnn: Towards balanced learning for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 821–830.
30. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37.
31. Li, Z.; Zhou, F. FSSD: Feature fusion single shot multibox detector. *arXiv* **2017**, arXiv:1712.00960.
32. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
33. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
34. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
35. Epshtein, B.; Ofek, E.; Wexler, Y. Detecting text in natural scenes with stroke width transform. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 2963–2970.
36. Matas, J.; Chum, O.; Urban, M.; Pajdla, T. Robust wide-baseline stereo from maximally stable extremal regions. *Image Vis. Comput.* **2004**, *22*, 761–767. [[CrossRef](#)]
37. Liao, M.; Shi, B.; Bai, X.; Wang, X.; Liu, W. Textboxes: A fast text detector with a single deep neural network. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.

38. Liu, Y.; Jin, L. Deep matching prior network: Toward tighter multi-oriented text detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1962–1969.
39. He, P.; Huang, W.; He, T.; Zhu, Q.; Qiao, Y.; Li, X. Single shot text detector with regional attention. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3047–3055.
40. Long, S.; Ruan, J.; Zhang, W.; He, X.; Wu, W.; Yao, C. Textsnake: A flexible representation for detecting text of arbitrary shapes. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 20–36.
41. Baek, Y.; Lee, B.; Han, D.; Yun, S.; Lee, H. Character region awareness for text detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–21 June 2019; pp. 9365–9374.
42. Lyu, P.; Liao, M.; Yao, C.; Wu, W.; Bai, X. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 67–83.
43. Everingham, M.; Eslami, S.A.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [[CrossRef](#)]
44. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Cham, Switzerland, 2014; pp. 740–755.
45. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 248–255.
46. Dutta, A.; Gupta, A.; Zissermann, A. VGG image annotator (VIA). 2016. Available online: <https://www.robots.ox.ac.uk/~vgg/software/via/> (accessed on 1 September 2020).
47. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
48. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.